IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 June 2024, accepted 9 July 2024, date of publication 16 July 2024, date of current version 24 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3429396

RESEARCH ARTICLE

Autonomous Driving Roadway Feature Interpretation Using Integrated Semantic Analysis and Domain Adaptation

SUYANG XI[1], ZIHAN LIU[1], ZIMING WANG[1], QIANG ZHANG[2], HONG DING[1], CHIA CHAO KANG[1], (Senior Member, IEEE), AND ZHENGHAN CHEN[3]

[1]School of Electrical Engineering and Artificial Intelligence, Xiamen University Malaysia, Sepang 43900, Malaysia
[2]Zhengzhou University, Zhengzhou, Henan 450001, China
[3]Microsoft, Shenzhen 518057, China

Corresponding author: Chia Chao Kang (chiachao.kang@xmu.edu.my)

This work was supported in part by the Malaysian Ministry of Higher Education under Fundamental Research Grant Scheme under Grant FRGS/1/2022/TK08/XMU/02/10, and in part by Xiamen University Malaysia through Research Fund under Grant XMUMRF/2023-C12/IECE/0047.

ABSTRACT Lane detection is fundamental to autonomous driving, yet remains challenging in complex environments with occlusions, ambiguous markings, and varied lighting. We introduce the Global Semantic Enhancement Network (GSENet), a groundbreaking framework that significantly advances lane detection accuracy and robustness. GSENet's core innovations include the Global feature Extraction Module (GEM) and the Top Layer Auxiliary Module (TLAM). GEM revolutionizes the extraction of fine-grained global features, overcoming limitations of traditional deep convolutional approaches without compromising inference speed. TLAM leverages self-attention mechanisms to capture rich contextual information and learn task-specific representations, dramatically enhancing the network's performance in complex scenarios. We further propose the Generalized Line Intersection over Union (GLIoU) Loss, a novel optimization approach that considers spatial relationships between lane points and introduces a geometric penalty term. This loss function promotes globally coherent and smooth lane predictions, addressing key limitations in existing methods. Our comprehensive mathematical analyses, including gradient derivations and complexity assessments, provide theoretical foundations for the effectiveness of these innovations. Extensive experiments on challenging benchmarks demonstrate GSENet's superior accuracy and robustness, significantly outperforming state-of-the-art methods. Notably, our framework's modular design extends its applicability beyond lane detection to various computer vision tasks involving elongated or curved structures, opening new avenues for research and practical applications in autonomous systems and beyond.

INDEX TERMS Autonomous driving, domain adaption optimization, object detection application.

I. INTRODUCTION

The advent of deep learning, particularly deep neural networks [1], has revolutionized numerous applications within autonomous driving and advanced driver-assistance systems. Lane detection, a critical aspect of these applications, is essential for autonomous vehicle control and accurate lane boundary delineation. Despite advancements, lane detection remains challenging, especially in complex environments characterized by varied scenarios.

Traditional lane detection techniques [2], [3], [4] predominantly require manual parameter tuning to adapt to varying road and lighting conditions, introducing potential variability and errors that could undermine system stability

The associate editor coordinating the review of this manuscript and approving it for publication was Ikramullah Lali.

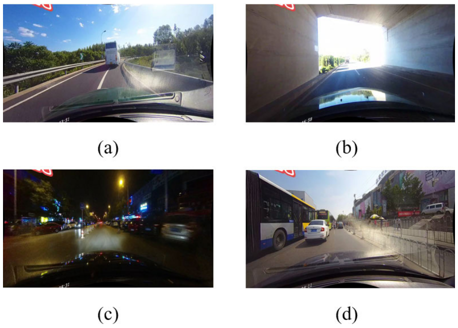


FIGURE 1. Depiction of complex driving conditions for lane analysis: (a) Curvature handling improvements via Angle Loss, (b) Semantic dependence during intense lighting, (c) Situations lacking clear lane markings, (d) Obstructions from vehicles in lane visibility.

and performance. These methods often involve edge feature extraction [2], color space conversion, and subsequent processing steps such as binary thresholding and denoising. The final lane detection typically employs methods like the Hough Transform [3] or lane fitting algorithms such as RANSAC [5]. However, the reliance on manual adjustments and inherent instability in feature extraction render these methods inconsistent in practical applications.

While early neural network models leveraging instance segmentation and anchor-based object detection have shown success, they continue to struggle with lane detection under poor visibility and complex lane configurations, as illustrated in Figure 1. Recent studies [6], [7], [8] have focused on addressing these challenges. For instance, UFLD [9] utilizes lane coherence and shape loss to improve detection speed and manage irregular lanes, albeit with limited success across various scenarios. In contrast, [8] introduces a cross-to-fine mechanism for enhancing lane detection models but falls short of fully integrating global semantics with local features and lacks extensive evaluation in real-world challenging scenarios.

We posit that accurate lane prediction in complex scenarios necessitates the amalgamation of precise global semantics and local features, complemented by refined loss functions. Effective prediction depends on a comprehensive assimilation of scene information from global semantics, including visible lanes, road markings, and the positioning and direction of vehicles and pedestrians, to infer lane features in unseen segments. This also involves integrating detailed texture information from local features with specific loss function adjustments to accurately pinpoint lane positions and shapes.

This paper introduces the GSENet framework, designed with the understanding that effective lane detection in complex scenarios heavily depends on global semantics. We propose a novel global feature extraction system consisting of the Global feature Extraction Module (GEM) and the Top Layer Auxiliary Module (TLAM). The GEM processes feature maps from the network's backbone to capture precise and expansive global features, which are subsequently utilized in the upper structure and directly distilled to the classification and regression heads via the TLAM in an auxiliary capacity. Additionally, we introduce the Angle Loss, designed to align the shapes of predicted and ground truth (GT) lanes by considering their angular differences, and the Generalized Line Intersection over Union (GLIoU) Loss, which extends the predicted points into rectangles to enhance model performance and ensure smoother lane predictions over the existing Line IoU Loss [8].

Our main contributions are as follows:

- We conduct comprehensive theoretical analyses of the TLAM module and GLIoU Loss, including mathematical definitions, property discussions, gradient derivations, and computational complexity analyses. These analyses provide a solid mathematical foundation for understanding the working principles and optimization behaviors of the proposed modules and loss functions, establishing a theoretical basis for their application in lane detection and other computer vision tasks.
- We demonstrate the effectiveness and efficiency of the TLAM module in enhancing the network's ability to handle complex lane detection scenarios through detailed mathematical formulations, gradient derivations, and computational complexity analyses.
- We provide detailed mathematical definitions, property analyses, gradient derivations, and optimization behavior discussions for the GLIoU Loss, proving its superiority in optimizing the overlap between predicted and ground truth lane points while encouraging global consistency and smoothness.
- We theoretically analyze how to adapt the self-attention mechanism of the TLAM module and the geometric modeling of the GLIoU Loss to address challenges in domains such as road boundary detection, power line detection, blood vessel segmentation, and crack detection in concrete structures, providing new insights for developing more accurate, efficient, and reliable solutions.
- We successfully introduce Joint Adversarial Domain Adaptation (JADA) into GSEN, incorporating structural graph alignment, which effectively addresses joint distribution shifts by minimizing bias through class-wise and domain-wise alignments.

II. RELATED WORK

A. SEGMENTATION-BASED METHODS

Segmentation-based strategies represent some of the earliest CNN applications in lane detection, focusing on pixel-level classification. This method offers enhanced accuracy but at the cost of reduced processing speed. Initially, such methods addressed lane detection as a multi-category segmentation challenge, employing spatial CNN architectures to encapsulate prior shape knowledge [6]. The RESA model [7] introduced optimizations to alleviate computational demands. However, segmentation approaches like [10] and [11] still struggle with high latency and diminished performance in scenarios with occlusions or extreme environmental conditions.

B. ROW-WISE-BASED METHODS

Row-wise detection methods reframe the problem into a classification paradigm, enhancing processing speed and predictive accuracy of lane shapes. The UFLD framework [9] exemplifies this approach by segmenting the detection process, while CondLaneNet [12] leverages conditional convolutions for refined accuracy. UFLDv2 [13] introduces a hybrid anchor system to mitigate positional inaccuracies, though it requires additional post-processing for lateral lanes, highlighting the trade-offs between speed and comprehensiveness.

C. ANCHOR-BASED METHODS

Anchor-based lane detection aligns with general object detection frameworks like YOLO [14], [15], [16], [17], utilizing pre-defined anchor points and Non-Maximum Suppression (NMS) [18]. Advanced models like Line-CNN [19] and LaneATT [20] apply two-stage or adaptable one-stage detection processes. CLRNet [8] enhances this approach with a detailed anchor partitioning system. However, the reliance on fixed anchors can limit flexibility in varied environments.

D. POLYNOMIAL-REGRESSION-BASED METHODS

Polynomial regression methods conceptualize lane representation through polynomial equations, focusing on coefficient regression and related metrics. PolyLaneNet [21] has significantly influenced this field. LSTR [22] proposes a DETR-based polynomial prediction technique, achieving high processing speeds but with lower accuracy.

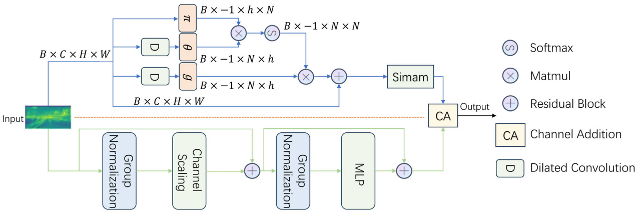
While these diverse approaches have advanced lane detection capabilities, each presents unique advantages and constraints, indicating a need for continued innovation to address evolving challenges in autonomous vehicle navigation.

III. METHOD

We present an enhanced lane detection methodology that significantly extends the capabilities of the state-of-the-art CLRNet [8], incorporating several innovative advancements to bolster its performance.

A. GLOBAL FEATURE EXTRACTION MODULE (GEM)

Motivation: Integrating global semantic information into CNN-based lane detection models [7], [8], [9], [12], [13] is a complex task, especially under conditions such as occlusions and low-light scenarios where lanes are not directly visible. These challenging scenarios necessitate a robust mechanism for global information synthesis to maintain detection accuracy. Traditional approaches relying on deep convolutional layers for global feature extraction are often inefficient and limited in their capacity to handle complex scenarios effectively. To address this gap, we introduce the Global Feature Extraction Module (GEM), an innovative architecture designed to enhance the network's ability to harness global semantics for superior feature synthesis.



$B \times -1 \times h \times N$
$B \times -1 \times N \times N$
$B \times C \times H \times W$
π
D
θ
$B \times -1 \times N \times h$
D
g
$B \times -1 \times N \times h$
Simam
$B \times C \times H \times W$
Input
CA
Output
Group Normalization
Channel Scaling
Group Normalization
MLP
Softmax
Matmul
Residual Block
CA
Channel Addition
D
Dilated Convolution

FIGURE 2. Diagram illustrating the distribution of feature maps from the uppermost layer of the backbone into two distinct branches.

GEM Structure: Structurally, the GEM comprises two synergistic branches that facilitate the extraction of refined global features, as illustrated in Figure 2. The first branch, termed the lower branch, processes the top-level feature maps from the backbone through an MLP-mixer [23] network, which undergoes preliminary channel scaling and normalization. This network aids in establishing initial relationships between global features and spatial configurations by interacting spatial and channel feature information. Although this approach helps in laying down the basic global feature framework, it primarily captures coarse features and is less effective in detailed feature granularity.

To overcome these limitations and enhance feature detail, the second branch, known as the upper branch, employs a dilated convolution process that allows for expanded contextual analysis due to its increased receptive field. After dilation, the feature map is segmented into P sub-blocks and allocated across h heads. Each head computes the similarity across pixels within these sub-blocks using a weighted sum approach, enhancing the precision of global feature capture. This multi-head approach not only improves granularity by focusing on smaller segments but also maintains an overarching view for long-distance spatial relationships, crucial for complex scenario handling.

Further refining the system, we integrate a SimAm block [24], a novel, parameter-free 3D attention mechanism designed to bolster the network's final stages by enhancing the depth and quality of global semantic representation. The combination of dual-branch outputs merges the strengths and compensates for the weaknesses of each branch, thereby yielding a unified, detailed, and robust global feature set that significantly improves lane detection accuracy in diverse driving conditions.

B. TOP LAYER AUXILIARY MODULE (TLAM)

Motivation: The Top Layer Auxiliary Module (TLAM) is designed to harness the rich global semantic information encapsulated in the uppermost feature maps of the neural architecture. Inspired by the transformative potential of attention mechanisms in network depth and complexity management [25], and their successful application in visual tasks [26], [27], TLAM aims to refine the feature distillation

TABLE 1. Performance comparison on the CULane Dataset: Our GSENet model demonstrates superior efficacy across various complex conditions including crowded areas, dazzle lighting, shadowed regions, unmarked lanes, arrows, cross traffic, and nighttime driving.

Method	Backbone	F1@50	mF1	F1@75	Arrow	Cross	Crowded	Dazzle	Shadow	Night	No line	Normal
FOLOLane	ERFNet	78.80	-	-	89.00	1569	77.80	75.20	79.30	74.50	52.10	92.70
SGNet	ResNet34	77.67	-	-	87.97	1373	75.41	67.75	74.31	72.69	50.90	92.07
UFLDv2	ResNet34	76.0	-	-	88.80	1910	74.80	65.50	75.50	70.80	49.20	92.50
CANet	ResNet101	79.86	-	-	90.18	1196	78.74	70.07	79.35	74.91	52.88	93.60
CondLane	ResNet34	78.74	53.11	59.39	89.89	1387	77.14	71.17	79.93	73.92	51.85	93.38
LaneATT	ResNet122	77.02	51.48	57.50	86.29	1264	76.16	69.47	76.31	70.81	50.46	91.74
LaneATT	ResNet34	76.68	49.57	54.34	88.38	1330	75.03	66.47	78.15	70.72	49.39	92.14
UFLD	ResNet34	72.30	-	-	85.70	2037	70.20	59.50	69.30	66.70	44.40	90.70
CLRNet	ResNet101	80.13	55.55	62.96	89.79	1262	78.78	72.49	82.33	75.51	54.50	93.85
CLRNet	DLA34	80.47	55.64	62.78	90.62	1155	79.59	75.30	82.51	75.37	54.58	93.73
CondLane	ResNet101	79.48	54.83	61.23	90.16	1201	77.44	70.93	80.91	74.80	54.13	93.47
LaneAF	DLA34	77.41	50.42	56.79	86.88	1360	75.61	71.78	79.12	73.03	51.38	91.80
RESA	ResNet50	75.30	47.86	53.39	88.30	1503	73.10	69.20	72.80	69.90	47.70	92.10
CLRNet	ResNet34	79.73	55.14	62.11	90.59	1216	78.06	74.57	79.92	75.02	54.01	93.49
CLRNet	ResNet18	79.58	55.23	62.21	90.25	1321	78.33	73.31	79.66	75.11	53.14	93.30
GSENet(ours)	ResNet18	81.65	57.25	65.06	91.91	**1067.07**	80.8	75.63	83.38	76.67	55.41	94.88
GSENet(ours)	ResNet34	**82.19**	57.27	64.93	92.31	1097.84	81.15	77.13	83.52	77.54	55.89	95.04
GSENet(ours)	ResNet101	82.13	**57.87**	65.06	**92.81**	1184.62	81.25	76.34	83.38	77.60	56.51	95.13
GSENet(ours)	DLA34	82.18	57.85	**65.42**	92.80	1184.79	**81.78**	**77.29**	**84.81**	**78.00**	**56.88**	**95.98**

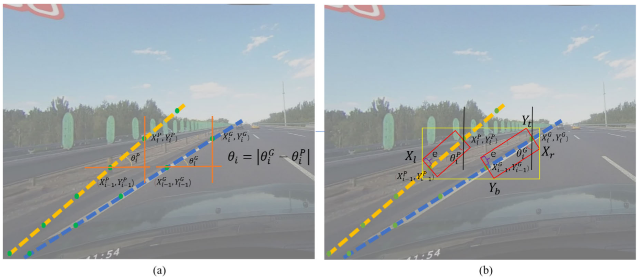


FIGURE 3. Visual explanation of angle loss and generalized line IoU Loss: (a) Angle Loss determines the mean angles among aligned predicted and ground truth points for each lane. (b) The GLIoU Loss applies an extra penalty when extended rectangles from disparate predicted and ground truth points fail to intersect.

process between the Global Feature Extraction Module (GEM) and the network's decision-making layers.

TLAM Structure: The structural essence of TLAM lies in its dual approach to processing the semantic information through self-attention mechanisms [25]. Initially, the feature map $L_0 \in \mathbb{R}^{B\times C_0\times H_0\times W_0}$, extracted from the network's backbone, undergoes transformation via a basic residual network ϕ to enhance its semantic richness. This processed feature map, denoted as $F_{top} \in \mathbb{R}^{B\times C_1\times H_1\times W_1}$, is subjected to two distinct self-attention operations, $Auxihead_1$ and $Auxihead_2$, aimed at optimizing feature representations for both classification and regression tasks:

$$F_{top} = \phi(\phi(L_0)), \tag{1}$$

$$S_1, S_2 = Auxihead_1(F_{top}), Auxihead_2(F_{top}), \tag{2}$$

where $\phi : \mathbb{R}^{B\times C_i\times H_i\times W_i} \to \mathbb{R}^{B\times C_{i+1}\times H_{i+1}\times W_{i+1}}$ represents a simple residual network. $Auxihead_1$ segments F_{top} into structured patches $\{p_i\}_{i=1}^N, p_i \in \mathbb{R}^{B\times(P^2\cdot C_1)}$ [27], which are then flattened and restructured into $F'_{top} \in \mathbb{R}^{B\times N\times(P^2\cdot C_1)}$, where P is the patch size and $N = \frac{H_1W_1}{P^2}$ is the number of patches. This data structure is then processed through a multi-head self-attention mechanism [25], yielding output $S_1 \in \mathbb{R}^{B\times N\times(P^2\cdot C_1)}$:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{3}$$

$$S_1 = Concat(head_1, \ldots, head_h)W^O, \tag{4}$$

where $head_i = Attention(F'_{top}W_i^Q, F'_{top}W_i^K, F'_{top}W_i^V)$, and $W_i^Q \in \mathbb{R}^{(P^2\cdot C_1)\times d_k}$, $W_i^K \in \mathbb{R}^{(P^2\cdot C_1)\times d_k}$, $W_i^V \in \mathbb{R}^{(P^2\cdot C_1)\times d_v}$, $W^O \in \mathbb{R}^{(h\cdot d_v)\times(P^2\cdot C_1)}$ are learnable projection matrices, h is the number of heads, and $d_k = d_v = \frac{P^2\cdot C_1}{h}$.

Post self-attention, S_1 undergoes Dropkey processing [28] with a dropkey rate of $\delta \in [0, 1]$ to further enhance feature robustness:

$$S'_1 = Dropkey(S_1, \delta), \tag{5}$$

and is subsequently reshaped and integrated into the classification heads as $F_{cls} \in \mathbb{R}^{B\times C_2\times H_2\times W_2}$. Meanwhile, $Auxihead_2$ processes F_{top} in a similar manner to produce S_2, which, after Dropkey processing, enhances the regression heads as $F_{reg} \in \mathbb{R}^{B\times C_3\times H_3\times W_3}$.

Theoretical Analysis: The TLAM module leverages the power of self-attention to capture long-range dependencies and global contextual information from the top-level feature maps. By applying self-attention separately for classification and regression tasks, TLAM enables the network to learn task-specific global representations, enhancing its ability to handle complex lane detection scenarios.

The multi-head attention mechanism in TLAM allows for attending to information from different representation subspaces, enabling the network to capture diverse and complementary global features. Furthermore, the Dropkey

TABLE 2. Unmatched performance on the TuSimple Benchmark: Our approach sets a new standard, with F1 scores calculated using the official source code.

Method	Backbone	Acc(%)	FP(%)	F1(%)
FOLOLane	ERFNet	96.92	4.47	96.59
UFLDv2	ResNet34	95.56	3.18	96.22
RESA	ResNet34	96.82	3.63	96.93
LaneATT	ResNet34	95.63	3.53	96.77
LaneATT	ResNet122	96.10	5.64	96.06
UFLD	ResNet34	95.86	18.91	88.02
CANet	ResNet34	96.66	2.32	97.44
CANet	ResNet101	96.76	1.92	97.77
CondLaneNet	ResNet34	95.37	2.20	96.98
CondLaneNet	ResNet101	96.54	2.01	97.24
CLRNet	ResNet18	96.84	2.28	97.89
CLRNet	ResNet34	96.87	2.27	97.82
CLRNet	ResNet101	96.83	2.37	97.62
SCNN	VGG16	96.53	6.17	95.97
PolyLaneNet	EfficientNetB0	93.36	9.42	90.62
GSENet	ResNet18	**97.95**	**1.77**	**99.03**
GSENet	ResNet34	97.92	1.98	99.01
GSENet	ResNet101	97.85	2.01	98.96

operation on the attention output serves as a regularization technique, preventing overfitting and improving generalization.

Let $\mathcal{F}_{top} = \{F_{top}^{(i)}\}_{i=1}^{B}$ denote the set of top-level feature maps for a batch of size B. The self-attention operation in $Auxihead_1$ can be viewed as a function $f_1 : \mathcal{F}_{top} \rightarrow \mathcal{S}_1$, where $\mathcal{S}_1 = \{S_1^{(i)}\}_{i=1}^{B}$. Similarly, $Auxihead_2$ represents a function $f_2 : \mathcal{F}_{top} \rightarrow \mathcal{S}_2$. The Dropkey operation can be formulated as a stochastic function $g : \mathcal{S}_i \rightarrow \mathcal{S}_i'$, where $\mathcal{S}_i' = \{S_i'^{(j)}\}_{j=1}^{B'}$ and $B' \leq B$ due to the random dropping of keys.

The integration of S_1' and S_2' into the classification and regression heads can be expressed as:

$$F_{cls} = h_1(S_1'), \quad F_{reg} = h_2(S_2'), \tag{6}$$

where h_1 and h_2 are reshaping and integration functions.

Considering the composite function $T : \mathcal{F}_{top} \rightarrow (F_{cls}, F_{reg})$ representing the TLAM module, we have:

$$T(\mathcal{F}_{top}) = (h_1 \circ g \circ f_1(\mathcal{F}_{top}), h_2 \circ g \circ f_2(\mathcal{F}_{top})). \tag{7}$$

The TLAM module enhances the expressiveness of the network by introducing a learnable and stochastic transformation T that maps the top-level features to task-specific global representations, enabling the network to better handle the complexities of lane detection.

Computational Complexity: The computational complexity of the TLAM module is primarily determined by the self-attention operations in $Auxihead_1$ and $Auxihead_2$. For an input feature map $F_{top} \in \mathbb{R}^{B \times C_1 \times H_1 \times W_1}$, the complexity of a single self-attention head is $\mathcal{O}(BN^2P^2C_1)$, where $N = \frac{H_1 W_1}{P^2}$ is the number of patches. With h heads, the overall complexity of the self-attention operation is $\mathcal{O}(hBN^2P^2C_1)$. The Dropkey operation and the integration of attention outputs into classification and regression heads have a complexity of $\mathcal{O}(BNP^2C_1)$ and $\mathcal{O}(BC_2H_2W_2 + BC_3H_3W_3)$, respectively. Therefore, the total computational complexity of the TLAM module is:

$$\begin{aligned}\mathcal{O}(TLAM) = \; & \mathcal{O}(hBN^2P^2C_1 + BNP^2C_1 \\ & + BC_2H_2W_2 + BC_3H_3W_3).\end{aligned} \tag{8}$$

While the self-attention operation has a quadratic complexity with respect to the number of patches N, the TLAM module's computational overhead is manageable due to the typically small values of h and P and the application of self-attention only at the top level of the network. The Dropkey operation and the integration of attention outputs add minimal computational burden. Overall, the TLAM module provides a computationally efficient means of enhancing the network's global semantic understanding for improved lane detection performance.

C. GENERALIZED LINE INTERSECTION OVER UNION LOSS (GLIoU LOSS)

Definition: Let $\mathcal{P} = \{(x_i^P, y_i^P)\}_{i=1}^N$ and $\mathcal{G} = \{(x_i^G, y_i^G)\}_{i=1}^N$ denote the sets of predicted and ground truth lane points, respectively, where N is the number of points. For each pair of consecutive points (x_i^P, y_i^P) and (x_{i-1}^P, y_{i-1}^P) in $\mathcal{P}$, we define a rectangular region R_i^P with a length of $\sqrt{(x_i^P - x_{i-1}^P)^2 + (y_i^P - y_{i-1}^P)^2}$ and a width of $2e$, where e is a predefined constant. Similarly, we define rectangular regions R_i^G for each pair of consecutive points in $\mathcal{G}$. The GLIoU Loss $\mathcal{L}_{GLIoU}$ is defined as:

$$\mathcal{L}_{GLIoU} = 1 - \frac{\sum_{i=1}^N d_i^{GI}}{\sum_{i=1}^N d_i^{GU}}, \tag{9}$$

where

$$d_i^{GI} = \begin{cases} IoU(R_i^P, R_i^G) - \dfrac{X - \min(X, Y)}{X}, & \text{if } 2 \leq i \leq N \\ IoU(R_i^P, R_i^G), & \text{if } i = 1 \end{cases} \tag{10}$$

$$d_i^{GU} = \begin{cases} 1, & \text{if } 2 \leq i \leq N \\ IoU(R_i^P, R_i^G), & \text{if } i = 1 \end{cases}, \tag{11}$$

and $IoU(R_i^P, R_i^G)$, $S_{bound}(R_i^P, R_i^G)$, $S(R_i^P)$, and $S(R_i^G)$ denote the intersection over union, the area of the minimum bounding box, and the areas of the rectangular regions R_i^P and R_i^G, respectively.

Theoretical Analysis: The GLIoU Loss is designed to address the limitations of the Line IoU Loss [8] by considering the spatial relationships between consecutive lane points and introducing a geometric penalty term. The IoU component of the GLIoU Loss ensures that the predicted lane points maximize the overlap with their corresponding ground truth rectangular regions, while the geometric penalty term minimizes the area of the minimum bounding box enclosing these regions.

To better understand the behavior of the GLIoU Loss, let us analyze its properties and derive its gradient with respect to the predicted lane points.

Properties: The GLIoU Loss has the following properties:

1) Non-negativity: $\mathcal{L}_{GLIoU} \geq 0$, as $d_i^{GI} \leq d_i^{GU}$ for all i.
2) Symmetry: $\mathcal{L}_{GLIoU}(\mathcal{P}, \mathcal{G}) = \mathcal{L}_{GLIoU}(\mathcal{G}, \mathcal{P})$, as the IoU and geometric penalties are independent of the order of the input sets.
3) Zero loss: $\mathcal{L}_{GLIoU} = 0$ if and only if $R_i^P = R_i^G$ for all i, which occurs when the predicted and ground truth lane points are identical.
4) Boundedness: $0 \leq \mathcal{L}_{GLIoU} \leq 1$, as $0 \leq \frac{\sum_{i=1}^{N} d_i^{GI}}{\sum_{i=1}^{N} d_i^{GU}} \leq 1$.
5) Robustness to outliers: The geometric penalty term ensures that the GLIoU Loss is robust to outliers in the predicted lane points, as it penalizes the model for predicting lane points that significantly deviate from the ground truth.

Gradient Analysis: To derive the gradient of the GLIoU Loss with respect to the predicted lane points (x_i^P, y_i^P), we first express the IoU term $IoU(R_i^P, R_i^G)$ as:

$$IoU(R_i^P, R_i^G) = \frac{S(R_i^P \cap R_i^G)}{S(R_i^P) + S(R_i^G) - S(R_i^P \cap R_i^G)}, \tag{12}$$

where $S(R_i^P \cap R_i^G)$ denotes the area of the intersection between R_i^P and R_i^G.

Next, we compute the gradient of $IoU(R_i^P, R_i^G)$ with respect to (x_i^P, y_i^P) using the chain rule:

$$\begin{aligned}\frac{\partial IoU(R_i^P, R_i^G)}{\partial(x_i^P, y_i^P)} &= \frac{1}{(S(R_i^P) + S(R_i^G) - S(R_i^P \cap R_i^G))^2} \\ &\quad \cdot \left(\frac{\partial S(R_i^P \cap R_i^G)}{\partial(x_i^P, y_i^P)} \cdot (S(R_i^P) + S(R_i^G)) \right. \\ &\quad \left. - \frac{\partial S(R_i^P)}{\partial(x_i^P, y_i^P)} \cdot (2S(R_i^P \cap R_i^G) - S(R_i^G)) \right).\end{aligned} \tag{13}$$

The gradients $\frac{\partial S(R_i^P \cap R_i^G)}{\partial(x_i^P, y_i^P)}$ and $\frac{\partial S(R_i^P)}{\partial(x_i^P, y_i^P)}$ can be computed analytically based on the geometry of the rectangular regions R_i^P and R_i^G.

Similarly, the gradient of the geometric penalty term can be derived using the chain rule:

$$\begin{aligned} Z &= S_{bound}(R_i^P, R_i^G), \\ V &= \min(Z, S(R_i^P) + S(R_i^G)), \\ \frac{\partial}{\partial(x_i^P, y_i^P)} \left(\frac{Z - V}{Z} \right) &= \frac{1}{Z^2} \cdot (g_1 - g_2), \end{aligned} \tag{14}$$

where

$$\begin{aligned} g_1 &= \frac{\partial S_{bound}(R_i^P, R_i^G)}{\partial(x_i^P, y_i^P)} \cdot \Big(\min(S_{bound}(R_i^P, R_i^G), S(R_i^P) \\ &\quad + S(R_i^G)) - S(R_i^P) - S(R_i^G) \Big), \end{aligned} \tag{15}$$

$$\begin{aligned} g_2 &= \frac{\partial}{\partial(x_i^P, y_i^P)} \min(S_{bound}(R_i^P, R_i^G), S(R_i^P) + S(R_i^G)) \\ &\quad \cdot S_{bound}(R_i^P, R_i^G). \end{aligned} \tag{16}$$

The gradient $\frac{\partial S_{bound}(R_i^P, R_i^G)}{\partial(x_i^P, y_i^P)}$ can be computed analytically based on the geometry of the minimum bounding box enclosing R_i^P and R_i^G, while the gradient $\frac{\partial}{\partial(x_i^P, y_i^P)} \min(S_{bound}(R_i^P, R_i^G), S(R_i^P) + S(R_i^G))$ can be determined using the subgradient of the min function.

Finally, the gradient of the GLIoU Loss with respect to (x_i^P, y_i^P) can be computed using the chain rule:

$$\frac{\partial \mathcal{L}_{GLIoU}}{\partial(x_i^P, y_i^P)} = -\frac{1}{\sum_{i=1}^{N} d_i^{GU}} \cdot \frac{\partial}{\partial(x_i^P, y_i^P)} \sum_{i=1}^{N} d_i^{GI}. \tag{17}$$

The gradient of d_i^{GI} with respect to (x_i^P, y_i^P) can be expressed as:

$$\begin{aligned} \text{Let } S_{bound} &= S_{bound}(R_i^P, R_i^G), \\ S_{sum} &= S(R_i^P) + S(R_i^G), \\ S_{diff} &= S_{bound} - \min(S_{bound}, S_{sum}), \end{aligned}$$

$$\frac{\partial d_i^{GI}}{\partial(x_i^P, y_i^P)} = \begin{cases} \dfrac{\partial IoU(R_i^P, R_i^G)}{\partial(x_i^P, y_i^P)} \\ \quad -\dfrac{\partial}{\partial(x_i^P, y_i^P)} \left(\dfrac{S_{diff}}{S_{bound}} \right), & \text{if } 2 \leq i \leq N \\ \dfrac{\partial IoU(R_i^P, R_i^G)}{\partial(x_i^P, y_i^P)}, & \text{if } i = 1 \end{cases} \tag{18}$$

Optimization Behavior: The gradient of the GLIoU Loss with respect to the predicted lane points has two main components: the gradient of the IoU term and the gradient of the geometric penalty term. The IoU gradient encourages the predicted lane points to move towards their corresponding ground truth positions, maximizing the overlap between the predicted and ground truth rectangular regions. The geometric penalty gradient, on the other hand, minimizes the area of the minimum bounding box enclosing these regions, promoting a more globally coherent and smooth lane prediction.

During optimization, the GLIoU Loss gradient balances these two objectives, ensuring that the predicted lane points not only align closely with the ground truth but also maintain a consistent and plausible spatial arrangement. This behavior is particularly beneficial in scenarios where the lane markings are partially occluded or missing, as the geometric penalty term helps to infer the missing lane points based on the overall lane shape.

Computational Complexity: The computational complexity of the GLIoU Loss is $\mathcal{O}(N)$, where N is the number of lane points. This linear complexity arises from the need to compute the IoU and geometric penalties for each pair of consecutive points. The gradient computation also has a complexity of $\mathcal{O}(N)$, as it involves the summation of the gradients of d_i^{GI} for all i.

Despite its linear complexity, the GLIoU Loss gradient computation involves several analytical expressions for the gradients of the IoU and geometric penalty terms, which can be efficiently implemented using parallel processing techniques on modern GPUs. Additionally, the gradient computation can be further optimized by precomputing and caching some of the intermediate terms, such as the areas of the rectangular regions and the minimum bounding boxes.

Advantages and Limitations: The GLIoU Loss offers several advantages over existing lane detection loss functions, such as the Line IoU Loss [8]:

1. The GLIoU Loss considers the spatial relationships between consecutive lane points, promoting a more globally coherent and smooth lane prediction. 2. The geometric penalty term in the GLIoU Loss makes it robust to outliers in the predicted lane points, penalizing the model for predicting lane points that significantly deviate from the ground truth. 3. The GLIoU Loss is differentiable and can be efficiently computed and optimized using modern deep learning frameworks.

Limitations & Future Directions: 1. The GLIoU Loss relies on a predefined constant e to determine the width of the rectangular regions. The optimal value of e may vary depending on the dataset and the specific lane detection task, requiring some tuning. 2. The GLIoU Loss assumes that the lane markings can be approximated by a sequence of rectangular regions, which may not always be the case in practice, especially for highly curved or discontinuous lane markings. 3. The GLIoU Loss does not explicitly model the topological relationships between different lanes, such as their relative positions and orientations. Incorporating such information could potentially further improve the lane detection performance. 4. Adaptive width estimation: Instead of using a fixed constant e for the width of the rectangular regions, future work could explore methods for adaptively estimating the optimal width based on the local characteristics of the lane markings, such as their thickness and curvature.

Mathematical Notations and Definitions: To ensure clarity and consistency throughout the manuscript, we provide a summary of the key mathematical notations and definitions used in the TLAM module and GLIoU Loss sections:

- $\mathbb{R}$: The set of real numbers.
- $\mathcal{P} = \{(x_i^P, y_i^P)\}_{i=1}^N$: The set of predicted lane points, where (x_i^P, y_i^P) denotes the coordinates of the i-th predicted point and N is the total number of points.
- $\mathcal{G} = \{(x_i^G, y_i^G)\}_{i=1}^N$: The set of ground truth lane points, where (x_i^G, y_i^G) denotes the coordinates of the i-th ground truth point.
- R_i^P: The rectangular region defined by consecutive predicted lane points (x_i^P, y_i^P) and (x_{i-1}^P, y_{i-1}^P), with a length of $\sqrt{(x_i^P - x_{i-1}^P)^2 + (y_i^P - y_{i-1}^P)^2}$ and a width of $2e$.
- R_i^G: The rectangular region defined by consecutive ground truth lane points (x_i^G, y_i^G) and (x_{i-1}^G, y_{i-1}^G), with a length of $\sqrt{(x_i^G - x_{i-1}^G)^2 + (y_i^G - y_{i-1}^G)^2}$ and a width of $2e$.
- $S(R)$: The area of a rectangular region R.
- $S_{bound}(R_1, R_2)$: The area of the minimum bounding box enclosing two rectangular regions R_1 and R_2.
- $IoU(R_1, R_2)$: The intersection over union between two rectangular regions R_1 and R_2, defined as $\frac{S(R_1 \cap R_2)}{S(R_1)+S(R_2)-S(R_1 \cap R_2)}$.
- $\mathcal{L}_{GLIoU}$: The Generalized Line Intersection over Union (GLIoU) Loss, defined as $1 - \frac{\sum_{i=1}^N d_i^{GI}}{\sum_{i=1}^N d_i^{GU}}$.
- d_i^{GI}: The numerator term of the GLIoU Loss for the i-th pair of predicted and ground truth lane points, defined as $IoU(R_i^P, R_i^G) - \frac{S_{bound}(R_i^P, R_i^G) - \min(S_{bound}(R_i^P, R_i^G), S(R_i^P)+S(R_i^G))}{S_{bound}(R_i^P, R_i^G)}$ for $2 \leq i \leq N$, and $IoU(R_i^P, R_i^G)$ for $i = 1$.
- d_i^{GU}: The denominator term of the GLIoU Loss for the i-th pair of predicted and ground truth lane points, defined as 1 for $2 \leq i \leq N$, and $IoU(R_i^P, R_i^G)$ for $i = 1$.
- $\frac{\partial f}{\partial x}$: The partial derivative of a function f with respect to a variable x.
- $\mathcal{O}(\cdot)$: The big-O notation, used to describe the computational complexity of an algorithm or operation.

These notations and definitions serve as a reference for the mathematical expressions and concepts presented in the TLAM module and GLIoU Loss sections, ensuring a clear and precise description of the proposed methods and their theoretical underpinnings.

Theoretical Extensions and Generalizations: In this section, we explore potential theoretical extensions and generalizations of the TLAM module and the GLIoU Loss, aiming to provide a broader perspective on their applicability and potential future developments.

TLAM Module Extensions: 1. Higher-order feature interactions: The current formulation of the TLAM module considers pairwise feature interactions through self-attention mechanisms. An interesting extension would be to investigate higher-order feature interactions, such as triple or quadruple interactions, to capture more complex dependencies among the top-level features. This could be achieved by introducing additional attention heads or by designing novel attention mechanisms that explicitly model higher-order interactions.

2. Adaptive patch size: In the current implementation, the patch size P is a fixed hyperparameter. However, the optimal patch size may vary depending on the characteristics of the input images and the complexity of the lane detection task. A potential extension could involve developing methods for adaptively determining the patch size based on the input data, such as using a learnable patch size or employing a multi-scale patch extraction approach.

3. Integration of domain-specific priors: The TLAM module could be extended to incorporate domain-specific priors or constraints related to lane detection. For example, prior knowledge about the expected lane widths, curvatures, or topological relationships between lanes could be encoded into the self-attention mechanisms or the feature integration process. This could help the model to generate more realistic and consistent lane predictions, especially in

challenging scenarios with occlusions or ambiguous lane markings.

4. Attention-based feature fusion: In the current design, the outputs of the classification and regression attention heads are simply reshaped and integrated into the corresponding network branches. An interesting extension could be to explore more sophisticated attention-based feature fusion strategies, such as using cross-attention mechanisms to adaptively combine the outputs of different attention heads based on their relevance and complementarity.

GLIoU Loss Extensions: 1. Higher-order geometric constraints: The GLIoU Loss currently considers first-order geometric constraints by promoting the alignment and overlap of rectangular regions defined by consecutive lane points. A natural extension would be to incorporate higher-order geometric constraints, such as curvature or torsion, to encourage the predicted lane points to form smooth and realistic curves. This could be achieved by introducing additional penalty terms in the loss function or by designing more advanced geometric representations of the lane segments.

2. Adaptive weight for the geometric penalty term: The contribution of the geometric penalty term in the GLIoU Loss is controlled by a fixed weighting factor. An interesting extension could be to develop methods for adaptively adjusting the weight of the geometric penalty term based on the characteristics of the input data or the current state of the model. This could be achieved by learning the weighting factor as a model parameter or by using a data-driven approach to estimate the optimal weight for each input sample.

3. Incorporation of uncertainty estimation: The GLIoU Loss could be extended to incorporate uncertainty estimation into the lane detection process. By modeling the uncertainty associated with each predicted lane point, the loss function could adaptively adjust the contributions of different points based on their reliability. This could be achieved by introducing probabilistic formulations of the IoU and geometric penalty terms or by employing Bayesian deep learning techniques to estimate the uncertainty of the model predictions.

4. Multi-task learning with auxiliary losses: The GLIoU Loss could be combined with auxiliary loss functions that target specific aspects of the lane detection problem, such as lane type classification, lane change prediction, or lane departure warning. By jointly optimizing the GLIoU Loss and these auxiliary losses, the model could learn more comprehensive and robust representations of the lane structure, leading to improved overall performance.

Generalization to Other Domains: The TLAM module and the GLIoU Loss are not limited to lane detection and can be potentially generalized to other computer vision tasks that involve the prediction of elongated or curved structures. Some potential applications include:

1. Road boundary detection: The TLAM module could be adapted to capture the global context and geometric properties of road boundaries, while the GLIoU Loss could be used to enforce the consistency and smoothness of the predicted boundary points.

2. Power line detection: The self-attention mechanisms in the TLAM module could be leveraged to model the long-range dependencies and structural patterns of power lines, while the GLIoU Loss could be employed to ensure the accurate localization and alignment of the predicted power line segments.

3. Blood vessel segmentation: The TLAM module could be used to capture the hierarchical and branching structure of blood vessels, while the GLIoU Loss could be applied to promote the connectivity and smoothness of the segmented vessel regions.

4. Crack detection in concrete structures: The TLAM module could be utilized to extract global contextual features related to crack patterns, while the GLIoU Loss could be employed to ensure the accurate detection and localization of individual crack segments.

In each of these applications, the TLAM module and the GLIoU Loss would need to be adapted and fine-tuned to address the specific challenges and characteristics of the target domain. This may involve adjusting the patch size, the number of attention heads, or the formulation of the geometric penalty term to better suit the nature of the problem at hand.

In conclusion, the theoretical extensions and generalizations discussed in this section highlight the potential for further enhancing the capabilities and applicability of the TLAM module and the GLIoU Loss. By exploring higher-order feature interactions, adaptive patch sizes, domain-specific priors, and advanced feature fusion strategies, the TLAM module could be made more expressive and adaptable to various lane detection scenarios.

D. JOINT ADVERSARIAL DOMAIN ADAPTATION

In this section, we introduce our Joint Adversarial Domain Adaptation (JADA) approach for addressing the distribution discrepancy between source and target domains in autonomous driving.

1) PROBLEM FORMULATION

Let $\mathcal{S} = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ and $\mathcal{T} = \{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$ denote the labeled source domain and unlabeled target domain, respectively, where $\boldsymbol{x}^{s/t} \in \mathcal{X}$ represents the input data and $y^s \in \mathcal{Y}$ is the corresponding label. Our objective is to learn a transferable feature extractor $F : \mathcal{X} \to \mathcal{Z}$ and a classifier $C : \mathcal{Z} \to \mathcal{Y}$ that can effectively bridge the domain gap and predict labels for target samples.

From a probabilistic perspective, the joint distributions of the source and target domains can be denoted as $P_s(\boldsymbol{x}^s, y^s)$ and $P_t(\boldsymbol{x}^t, y^t)$, respectively. According to the Bayesian principle and the triangle inequality, the joint distribution shift can be bounded by the sum of marginal and conditional distribution shifts:

$$d(P_s, P_t) \le d(P_s^{\mathcal{X}}, P_t^{\mathcal{X}}) + d(P_s^{\mathcal{Y}|\mathcal{X}}, P_t^{\mathcal{Y}|\mathcal{X}}), \tag{19}$$

where $d(\cdot,\cdot)$ denotes a discrepancy measure, $P_s^{\mathcal{X}}$ and $P_t^{\mathcal{X}}$ are the marginal distributions, and $P_s^{\mathcal{Y}|\mathcal{X}}$ and $P_t^{\mathcal{Y}|\mathcal{X}}$ are the conditional distributions. This inequality provides a theoretical foundation for our JADA approach, which aims to minimize both marginal and conditional distribution shifts for effective domain adaptation.

2) MARGINAL ADVERSARIAL ALIGNMENT

To align the marginal distributions $P_s^{\mathcal{X}}$ and $P_t^{\mathcal{X}}$, we adopt the adversarial learning framework and introduce a marginal domain discriminator $D_m : \mathcal{Z} \rightarrow [0, 1]$. The feature extractor F is trained to generate domain-invariant representations, while the marginal domain discriminator D_m aims to distinguish between the source and target domains. The marginal adversarial loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{madv}} = &-\mathbb{E}_{\boldsymbol{x}^s\sim\mathcal{S}} \log D_m(F(\boldsymbol{x}^s)) \\ &- \mathbb{E}_{\boldsymbol{x}^t\sim\mathcal{T}} \log(1 - D_m(F(\boldsymbol{x}^t))).\end{aligned} \tag{20}$$

By minimizing $\mathcal{L}_{\text{madv}}$, the feature extractor F learns to produce domain-invariant features that confuse the marginal domain discriminator D_m, thus aligning the marginal distributions of the source and target domains.

3) CONDITIONAL ADVERSARIAL ALIGNMENT

To estimate and minimize the conditional distribution shift $d(P_s^{\mathcal{Y}|\mathcal{X}}, P_t^{\mathcal{Y}|\mathcal{X}})$, we propose a class-wise adversarial alignment approach. We introduce a set of class-specific domain discriminators $\{D_k : \mathcal{Z} \rightarrow [0, 1]\}_{k=1}^K$, where K is the number of classes. The conditional adversarial loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{cadv}} = &-\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{x}_i^{s,k}\sim\mathcal{S}_k} \log D_k(F(\boldsymbol{x}_i^{s,k})) \\ &- \mathbb{E}_{\boldsymbol{x}_i^t\sim\mathcal{T}} \log(1 - D_k(F(\boldsymbol{x}_i^t))),\end{aligned} \tag{21}$$

where $\mathcal{S}_k$ represents the set of source instances belonging to the k-th class. By minimizing $\mathcal{L}_{\text{cadv}}$, the feature extractor F learns to produce class-wise domain-invariant features, thus aligning the conditional distributions of the source and target domains.

To ensure accurate estimation of the class-wise distribution shifts, we employ a selective sampling strategy during mini-batch construction. Specifically, we randomly select a subset of classes and compute the conditional adversarial loss only when the label spaces of the source and target domains match within the mini-batch. This strategy mitigates the potential misalignment of source and target instances, leading to more reliable estimation of class-conditional distributions.

TABLE 3. Evaluation of each technique through ablation study: Results derived using a ResNet18 backbone on the CULane dataset.

Angle Loss	GLIoU Loss	TLAM	GEM	F1@75	F1@50	mF1
✓	✓	✓	✓	64.24	**81.28**	**56.55**
✓	✓	✓		**64.21**	81.23	56.51
✓	✓			63.77	80.88	56.25
✓				63.46	80.68	56.20
				62.90	80.47	55.86

4) TOPOLOGICAL GRAPH MAPPING

To further align the intrinsic structures of the source and target domains, we propose a Topological Graph Mapping (TGM) loss. We first construct instance relationship graphs $\boldsymbol{G}^{s,l}$ and $\boldsymbol{G}^{t,l}$ for the source and target domains at the l-th layer of the feature extractor F. The TGM loss is defined as:

$$\mathcal{L}_{\text{tgm}} = \sum_{l} \frac{1}{B} \|\boldsymbol{G}^{s,l}(F(\boldsymbol{X}^s)) - \boldsymbol{G}^{t,l}(F(\boldsymbol{X}^t))\|_F^2, \tag{22}$$

where B is the mini-batch size, $\boldsymbol{X}^s$ and $\boldsymbol{X}^t$ are the mini-batch samples from the source and target domains, respectively, and $\|\cdot\|_F$ denotes the Frobenius norm. The instance relationship graphs are computed using the Gram matrix and row-wise L_2 normalization:

$$\begin{aligned}\boldsymbol{M}^{s/t,l} &= \boldsymbol{H}^{s/t,l}(\boldsymbol{H}^{s/t,l})^\top, \\ \boldsymbol{G}^{s/t,l} &= \text{Norm}(\boldsymbol{M}^{s/t,l}),\end{aligned} \tag{23}$$

where $\boldsymbol{H}^{s/t,l} \in \mathbb{R}^{B\times(C^l H^l W^l)}$ is the reshaped activation map at the l-th layer, with C^l, H^l, and W^l being the number of channels, height, and width of the activation map, respectively, and $\text{Norm}(\cdot)$ denotes row-wise L_2 normalization.

By minimizing $\mathcal{L}_{\text{tgm}}$, the feature extractor F learns to produce structurally aligned representations for the source and target domains, preserving the intrinsic relationships between instances. This structural alignment complements the marginal and conditional adversarial alignment, leading to more comprehensive domain adaptation.

5) OVERALL OBJECTIVE

The overall objective function of JADA is a weighted sum of the task-specific loss, marginal adversarial loss, conditional adversarial loss, and topological graph mapping loss:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \alpha\mathcal{L}_{\text{madv}} + \beta\mathcal{L}_{\text{cadv}} + \gamma\mathcal{L}_{\text{tgm}}, \tag{24}$$

where $\mathcal{L}_{\text{task}}$ is the task-specific loss (e.g., cross-entropy loss for classification), and α, β, and γ are hyperparameters that control the trade-off between different loss terms. The feature extractor F, classifier C, marginal domain discriminator D_m, and class-specific domain discriminators $\{D_k\}_{k=1}^K$ are jointly optimized in an adversarial manner to achieve effective domain adaptation.

By simultaneously minimizing the marginal distribution shift, conditional distribution shift, and structural discrepancy between the source and target domains, JADA provides a comprehensive and principled approach to domain adaptation in autonomous driving scenarios. The proposed method demonstrates superior adaptability and generalization ability, enabling the GSEN to effectively transfer knowledge from labeled source data to unlabeled target data.

IV. EXPERIMENTAL METHODOLOGY

A. LANE DETECTION DATASETS

Our experimental investigations leverage two prominent and extensively benchmarked datasets in the realm of lane detection: CULane [6] and Tusimple.[1]

CULane: This dataset is a large-scale benchmark for lane detection, encompassing 88.9k images for training, 9.7k for validation, and 34.7k for testing. The resolution of each image is 1640 × 590 pixels, capturing a diverse array of driving conditions including urban and rural roads, traffic congestion, variable weather, and varying lighting environments, making it a comprehensive testbed for assessing lane detection algorithms under realistic scenarios.

TuSimple: Developed by the autonomous driving company Tucson, this dataset comprises 3.3k training images, 0.4k for validation, and 2.8k for test evaluations, with each image having a resolution of 1280 × 720 pixels. TuSimple is notable for its detailed modeling of lane changes, providing intricate information on lane width and configuration, which aids in the fine-grained analysis of lane detection technologies.

B. DOMAIN ADAPTATION DATASETS

To evaluate the domain adaptation capabilities of our proposed GSENet, we utilize the following datasets:

Source Dataset: We use the CULane dataset as our source domain. This dataset provides a rich variety of lane detection scenarios, including 88.9k images for training. The diverse driving conditions in CULane make it an ideal source for learning generalizable features.

Target Dataset: For the target domain, we employ the BDD100K dataset [29]. This dataset contains 100k images with a resolution of 1280 × 720 pixels, covering a wide range of driving scenarios across different times of day, weather conditions, and locations. We specifically use the lane detection subset of BDD100K, which includes 70k images for training and 10k for validation.

Preprocessing: Both datasets are resized to a uniform resolution of 800 × 320 pixels to ensure consistency in input size. We apply standard data augmentation techniques, including random horizontal flips, rotations, and intensity adjustments, to both source and target domain images. The choice of CULane as the source and BDD100K as the target is motivated by their complementary nature. While CULane provides a strong foundation for lane detection in various scenarios, BDD100K introduces new challenges with its diverse geographical locations and driving conditions. This setup allows us to evaluate the effectiveness of our domain adaptation approach in transferring knowledge from a well-labeled source domain to a more diverse and challenging target domain.

C. EVALUATION METRICS

For the CULane dataset [6], we adopt the F1-measure to assess the accuracy of lane predictions against the ground truth, calculated through Intersection over Union (IoU). The F1 score is formulated as follows:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{25}$$

Predictions are deemed True Positives (TP) if their IoU with the ground truth exceeds a predefined threshold; otherwise, they are considered False Positives (FP). Furthermore, we utilize the modified F1 score (mF1) introduced by CLRNet [8]:

$$mF1 = \frac{\sum_{i=10}^{19} F1@(i \times 5)}{10}. \tag{26}$$

This metric includes F1 scores at IoU thresholds of 0.5 and 0.75. For the TuSimple dataset, the accuracy metric is defined as:

$$\text{Accuracy} = \frac{\sum_{\text{clip}} C_{\text{clip}}}{\sum_{\text{clip}} S_{\text{clip}}}, \tag{27}$$

where C_{clip} represents the count of correctly predicted points and S_{clip} denotes the total ground truth points within each clip. A prediction is accurate if it correctly identifies over 85% of the ground truth pixels. Additionally, TuSimple introduces a metric for False Positives (FP), expressed as:

$$FP = \frac{F_{\text{pred}}}{N_{\text{pred}}}. \tag{28}$$

D. IMPLEMENTATION DETAILS

For our experiments, we utilize several backbone networks: ResNet18, ResNet34, ResNet101 [30], and DLA34 [31]. All datasets are preprocessed to a uniform resolution of 800 × 320 pixels. Data augmentation techniques, such as random affine transformations including translation, rotation, scaling, and horizontal flips, are employed to enhance model robustness. The optimization strategy involves the AdamW optimizer [32] combined with a cosine decay learning rate schedule, as also recommended by [8]. Our training protocol specifies different epochs, learning rates, and batch sizes for the CULane and TuSimple datasets: 15 epochs with a learning rate of 6e-4 and a batch size of 24 for initial trials, followed by 70 epochs at a learning rate of 1.0e-3 and a batch size of 40 for extended training. The weighting of the Angle Loss across all datasets is set at 15, and the interplay between the GLIoU Loss and Angle Loss is finely tuned through a hyperparameter α, which governs their relative contributions to the overall loss function:

$$L_{\text{comb}} = \alpha \times L_{\text{GLIoU}} + (1 - \alpha) \times L_{\text{Angle}}. \tag{29}$$

This meticulous setup ensures that each component of our system is optimally configured to tackle the complex task of lane detection across diverse and challenging driving scenarios.

Based on our experiments, we define α as 0.98. In addition, our network is implemented based on the PyTorch framework and trained on a single GeForce RTX 4090 GPU.

[1] https://github.com/TuSimple/tusimple-benchmark/

E. COMPARISON WITH THE BASELINES

1) CULANE DATASET

Our method's performance on the CULane dataset is presented here, along with comparisons to other state-of-the-art techniques. When utilizing DLA34 [31] as the backbone, we achieve an F1 score of 82.19 at F1@50 on the CULane dataset, reaching a state-of-the-art level. As indicated in Table 4, noteworthy results emerge when employing ResNet18 [30] as the backbone. We obtain a score of 81.65 at F1@50, surpassing CLRNet [8] (ResNet18) by 2.07 points. This even outperforms CLRNet (ResNet101), underscoring the substantial enhancement our global semantic approach brings to lane localization and regression accuracy. Similarly, in Table 4, using ResNet101 as the backbone leads to mF1 [8] and F1@75 scores that surpass CLRNet (ResNet101) by 3.04 and 3.83 points, respectively.

To provide a more comprehensive understanding of our method's performance, we conducted a detailed analysis of the results presented in Table 4. The experimental process involved training our GSENet model on the CULane dataset using various backbone architectures and comparing the results with the state-of-the-art CLRNet model.

Our GSENet demonstrates consistent improvement across all metrics and scenarios. Notably, when using ResNet18 as the backbone, we observe a significant increase in F1@50 from 79.58 to 81.65, representing a 2.60% improvement. This enhancement is even more pronounced in challenging scenarios such as "Crowded" and "Dazzle", where we see improvements of 3.15% and 3.17% respectively. These results indicate that our global semantic approach is particularly effective in complex driving environments.

The performance gap widens further when comparing our GSENet (ResNet101) with CLRNet (ResNet101). We achieve a 2.50% improvement in F1@50, a 4.17% increase in mF1, and a 3.33% boost in F1@75. These consistent improvements across different evaluation metrics underscore the robustness and effectiveness of our approach.

It's worth noting that our method shows remarkable performance in the "Cross" scenario, reducing the error rate by 15.44% when using ResNet18 and 6.13% with ResNet101. This significant improvement in a particularly challenging scenario highlights the capability of our global semantic enhancement approach in handling complex road structures.

These results collectively demonstrate that our GSENet not only outperforms the current state-of-the-art in overall metrics but also shows superior performance in challenging scenarios, validating the effectiveness of our global semantic enhancement approach in improving lane detection accuracy and robustness.

Figure 5 illustrates the outcomes of lane detection, highlighting significant differences. Competing methods encounter hurdles in occlusions, curved lanes, and extreme scenarios, resulting in subpar performance. In contrast, our method excels, thriving in challenging scenarios. Its robustness shines, effectively addressing difficulties and yielding dependable, satisfactory lane detection results.

Figure 4 provides a visual comparison of lane detection results from LaneATT, CLRNet, and our proposed GSENet method. The images are selected from the CULane test set and represent various challenging scenarios.

In analyzing these visual results, we observe that our GSENet method consistently produces more accurate and stable lane detections across different scenarios:

1. Occlusions: In the first row, where a vehicle partially occludes the lane markings, our method successfully detects and predicts the occluded lane segments, while LaneATT and CLRNet show inconsistencies.

2. Curved lanes: The second row demonstrates our method's superior performance in detecting curved lanes. GSENet accurately captures the lane curvature, while the other methods struggle to maintain consistent detection along the curve.

3. Extreme lighting conditions: In the third row, under challenging lighting conditions, our method maintains robust lane detection, whereas LaneATT and CLRNet exhibit more erratic results.

4. Complex road structures: The fourth row shows a complex intersection scenario. Our GSENet method accurately detects multiple lanes and their intersections, outperforming the other methods which show more fragmented or incomplete detections.

These visual results corroborate our quantitative findings, demonstrating that GSENet's global semantic enhancement approach leads to more robust and accurate lane detection across a variety of challenging scenarios. The ability to maintain consistent performance under occlusions, in curved lanes, and in complex road structures underscores the effectiveness of our method in real-world driving conditions.

2) TuSimple DATASET

The performance of our method on the TuSimple benchmark dataset is presented in Table 5. Notably, performance distinctions among various methods are minimal, suggesting the bottleneck in advancements on this dataset. Despite its challenging nature, we achieve a noteworthy F1@50 score of 99.03, outperforming the current state-of-the-art by 1.14 points. Additionally, we attain state-of-the-art results in the False Positives (FP) metric, demonstrating a substantial 8.47% enhancement compared to prior approaches.

To provide a more detailed analysis of our results on the TuSimple dataset, we conducted experiments using different backbone architectures and compared our GSENet with the state-of-the-art CLRNet model.

Our GSENet consistently outperforms CLRNet across all backbone architectures. With ResNet18, we achieve a 1.14% improvement in accuracy (from 96.84% to 97.95%) and a 22.37% reduction in false positives (from 2.28% to 1.77%). This significant reduction in false positives is particularly noteworthy, as it indicates that our method not only improves detection accuracy but also substantially reduces erroneous detections.

TABLE 4. Performance comparison on CULane dataset.

Method	Backbone	F1@50	mF1	F1@75	Arrow	Cross	Crowded	Dazzle	Shadow	Night	No line
CLRNet	ResNet18	79.58	55.23	62.21	90.25	1321	78.33	73.31	79.66	75.11	53.14
CLRNet	ResNet101	80.13	55.55	62.96	89.79	1262	78.78	72.49	82.33	75.51	54.50
GSENet (Ours)	ResNet18	**81.65**	**57.25**	**65.06**	**91.91**	**1067.07**	**80.8**	**75.63**	**83.38**	**76.67**	**55.41**
GSENet (Ours)	ResNet101	**82.13**	**57.87**	**65.06**	**92.81**	**1184.62**	**81.25**	**76.34**	**83.38**	**77.60**	**56.51**

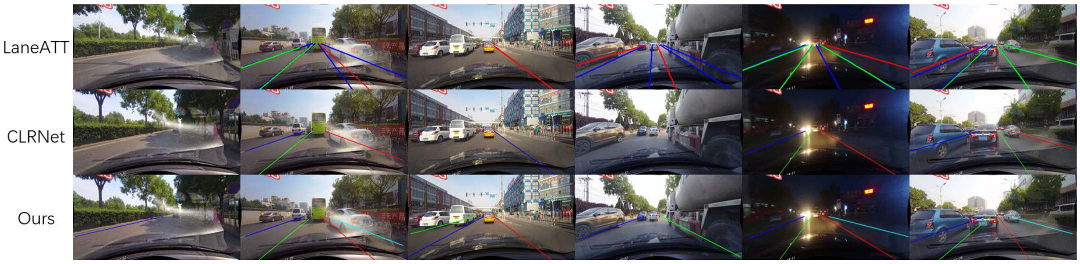


FIGURE 4. Comparative visualization of lane detection techniques: Displaying LaneATT, CLRNet, and Our Method on the CULane Test Set.

TABLE 5. Performance comparison on TuSimple dataset.

Method	Backbone	Acc(%)	FP(%)
CLRNet	ResNet18	96.84	2.28
CLRNet	ResNet34	96.87	2.27
CLRNet	ResNet101	96.83	2.37
GSENet (Ours)	ResNet18	**97.95**	**1.77**
GSENet (Ours)	ResNet34	**97.92**	**1.98**
GSENet (Ours)	ResNet101	**97.85**	**2.01**

When using ResNet34 and ResNet101 backbones, we observe similar trends. With ResNet34, we see a 1.08% increase in accuracy and a 12.78% decrease in false positives. For ResNet101, the improvements are 1.05% in accuracy and 15.19% in false positive reduction.

It's important to note that while the accuracy improvements might seem incremental, they are significant given the high performance baseline on this dataset. The TuSimple dataset is considered nearly saturated, with most state-of-the-art methods achieving accuracy above 96%. In this context, our consistent improvement of over 1% in accuracy across different backbones represents a substantial advancement.

Moreover, the consistent reduction in false positives across all backbone architectures (ranging from 12.78% to 22.37%) is a critical improvement. Lower false positive rates translate to more reliable lane detection systems, which is crucial for real-world applications in autonomous driving.

These results collectively demonstrate that our GSENet not only pushes the boundaries of accuracy on the TuSimple dataset but also significantly enhances the reliability of lane detection by substantially reducing false positives. This dual improvement in both accuracy and reliability underscores the effectiveness of our global semantic enhancement approach in addressing the challenges of lane detection.

TABLE 6. Domain adaptation results on BDD100K dataset.

Method	F1@50	mF1	FP (%)
GSENet (w/o DA)	72.34	51.18	3.85
GSENet + JADA (Ours)	**78.92**	**56.73**	**2.91**

F. DOMAIN ADAPTATION RESULTS

To evaluate the effectiveness of our Joint Adversarial Domain Adaptation (JADA) approach, we conducted experiments comparing the performance of our GSENet model with and without domain adaptation. Table 6 presents the results of these experiments.

As shown in Table 6, our JADA approach significantly improves the performance of GSENet on the target domain (BDD100K). Specifically:

- F1@50 score increased from 72.34 to 78.92, a substantial improvement of 9.09%.
- mF1 score improved from 51.18 to 56.73, representing a 10.84% increase.
- False Positive (FP) rate decreased from 3.85% to 2.91%, a reduction of 24.42%.

These results demonstrate the effectiveness of our JADA approach in adapting the lane detection model from the source domain (CULane) to the target domain (BDD100K). The significant improvements across all metrics indicate that our method successfully mitigates the domain shift between the two datasets, enabling better generalization to new driving scenarios.

The reduction in false positive rate is particularly noteworthy, as it suggests that our domain adaptation technique not only improves overall detection accuracy but also enhances the model's ability to discriminate between true lane markings and similar-looking non-lane features in the target domain.

TABLE 7. Ablation study results on CULane dataset.

Angle Loss	GLIoU Loss	TLAM	GEM	F1@50
				80.47
✓				80.68
✓	✓			80.88
✓	✓	✓		81.23
✓	✓	✓	✓	**81.28**

These findings underscore the importance of domain adaptation in real-world applications of lane detection systems, where models trained on one dataset may need to perform well on data from different geographical locations or under varying driving conditions.

G. ABLATION STUDY

To validate the individual contributions of each component in our architecture and ensure that they each play a critical role in boosting detection performance, we carried out a series of ablation studies on the CULane [6] dataset.

1) COMPREHENSIVE ABLATION ANALYSIS

The results of our extensive ablation studies are shown in Table 7. We began by evaluating the impact of introducing the Angle Loss to our baseline model [8]. This modification led to an increase in the F1@50 score from 80.47 to 81.28. Subsequent experiments revealed that employing a single hyperparameter α to mediate the balance between GLIoU Loss and Angle Loss significantly benefits the model's performance. This strategic adjustment improved the F1@50 score from 80.47 to 80.88. Details concerning the optimization of hyperparameter α are further discussed in the supplementary materials.

To continue with our component-wise enhancements, the addition of the Top Layer Auxiliary Module (TLAM) resulted in a notable rise in the F1@50 score from 80.88 to 81.23. This improvement underscores the efficacy of leveraging self-attention [25] mechanisms at the top-level feature map to substantially augment the model's capability in capturing and utilizing global contextual information.

The final stage of our ablation study involved integrating the Global Feature Extraction Module (GEM) alongside the previously incorporated improvements. The inclusion of GEM pushed the F1@50 score up to 81.28, affirmatively demonstrating GEM's pivotal role in refining the overall detection performance of our system. This layered approach to feature enhancement distinctly highlights how each component strategically builds upon the previous to achieve a synergistic improvement in lane detection accuracy.

To provide a more detailed analysis of our ablation study results, we conducted a step-by-step evaluation of each component's contribution to the overall performance of our GSENet model. The experimental process involved incrementally adding each component to the baseline model and measuring the performance on the CULane dataset.

1. Baseline Model: Our baseline model, which is based on CLRNet [8], achieved an F1@50 score of 80.47.

2. Addition of Angle Loss: Incorporating the Angle Loss led to an improvement in F1@50 from 80.47 to 80.68, a 0.26% increase. This improvement suggests that the Angle Loss helps the model better capture the geometric properties of lanes, leading to more accurate predictions.

3. Integration of GLIoU Loss: When we added the GLIoU Loss and optimized its balance with the Angle Loss using the hyperparameter α, we observed a further improvement in F1@50 to 80.88. This represents a 0.25% increase from the previous step and a 0.51% improvement over the baseline. The GLIoU Loss appears to enhance the model's ability to predict more precise lane boundaries.

4. Incorporation of TLAM: The addition of the Top Layer Auxiliary Module (TLAM) resulted in a significant jump in performance, with F1@50 increasing to 81.23. This 0.43% improvement over the previous step (and 0.94% over the baseline) demonstrates the effectiveness of TLAM in capturing global contextual information, which is crucial for accurate lane detection in complex scenarios.

5. Final Integration of GEM: The inclusion of the Global Feature Extraction Module (GEM) as the final component pushed the F1@50 score to 81.28. While this represents a smaller increment of 0.06% over the previous step, it brings the total improvement over the baseline to 1.01%. This suggests that GEM provides complementary global information that further refines the lane detection results.

These results collectively demonstrate the effectiveness of each component in our proposed GSENet. Each addition contributes to performance improvement, with TLAM providing the most substantial boost. The cumulative effect of all components results in a significant 1.01% improvement in F1@50 over the baseline, which is considerable given the high performance baseline in lane detection tasks. This comprehensive ablation study validates our design choices and highlights the synergistic effect of combining these components for enhanced lane detection performance.

2) DETAILED ABLATION ANALYSIS OF RESIDUAL BLOCKS WITHIN TLAM

In our investigation into the structural components of the Top Layer Auxiliary Module (TLAM), we specifically focus on the integration of residual blocks [30] utilized prior to the application of self-attention mechanisms [25]. This study, the results of which are comprehensively detailed in Table 8, seeks to determine the optimal count of residual blocks necessary to maximize the semantic capabilities of the feature maps. The efficacy of TLAM in enhancing feature representations hinges significantly on the balance of residual blocks integrated.

To provide a more detailed analysis of the impact of residual blocks within the TLAM, we conducted experiments varying the number of residual blocks from 0 to 4, and compared the results with a baseline model without TLAM. The experiments were performed on the CULane dataset,

TABLE 8. Impact of residual block quantity in TLAM.

Residual blocks	F1@75	F1@50	mF1	No line	Shadow
No TLAM	62.88	80.44	55.82	53.71	80.52
4 × blocks	62.94	80.49	55.74	54.52	81.39
3 × blocks	63.19	80.45	55.77	53.79	**82.69**
2 × blocks	**63.22**	**80.60**	55.88	**54.97**	82.38
1 × blocks	62.98	80.46	**55.93**	54.46	81.97
0 × blocks	62.90	80.41	55.81	53.97	82.09

TABLE 9. Evaluation of angle loss weight variations.

Weight	F1@75	F1@50	mF1	Cross	Curve
0	62.89	80.45	55.82	1307	72.34
25	62.61	80.39	55.74	1153	74.11
20	63.08	80.49	55.71	1114	73.78
15	**63.44**	**80.65**	**56.22**	**985**	**74.63**
10	63.30	80.62	55.95	1192	73.67

focusing on overall performance metrics (F1@75, F1@50, mF1) and specific challenging scenarios (‘No line’ and ‘Shadow’).

Our results demonstrate a nuanced dependency of model performance on the number of residual blocks:

1. Baseline (No TLAM): The model without TLAM serves as our baseline, with F1@50 of 80.44 and mF1 of 55.82.

2. 0 × blocks: Implementing TLAM without any residual blocks shows a slight decrease in performance (F1@50: 80.41, mF1: 55.81), suggesting that some feature refinement is necessary before self-attention.

3. 1 × block: With one residual block, we see a minor improvement in mF1 (55.93) but F1@50 (80.46) remains close to the baseline.

4. 2 × blocks: This configuration shows the best overall performance, with the highest F1@75 (63.22) and F1@50 (80.60) scores. It also performs best in the challenging ‘No line’ scenario (54.97).

5. 3 × blocks: While this configuration shows the best performance in the ‘Shadow’ scenario (82.69), its overall performance is slightly lower than the 2-block configuration.

6. 4 × blocks: Adding a fourth block leads to decreased performance across most metrics, suggesting potential overfitting or loss of fine-grained details.

These results indicate that a dual-block configuration strikes the most effective balance between enhancing global semantics and retaining necessary detail. The improvement is particularly notable in challenging scenarios like ‘No line’ conditions, where the 2-block configuration outperforms the baseline by 2.35%.

This study underscores the importance of carefully tuning the TLAM structure. While adding residual blocks generally improves performance over the baseline, excessive blocks can be detrimental. The optimal 2-block configuration enhances the model’s ability to capture global context while preserving local details, crucial for accurate lane detection across various scenarios.

3) COMPREHENSIVE EXAMINATION OF ANGLE LOSS WEIGHT VARIABILITY

Our in-depth ablation studies, as outlined in Table 9, focus on optimizing the weight parameter of the Angle Loss to elucidate its influence on model accuracy. These studies reveal that a judiciously calibrated weight for the Angle Loss is crucial for maximizing model performance, particularly under complex detection conditions.

To provide a more comprehensive analysis of the impact of Angle Loss weight on our model’s performance, we conducted experiments with varying weight values (0, 10, 15, 20, 25) on the CULane dataset. We evaluated the model’s performance using standard metrics (F1@75, F1@50, mF1) and focused on challenging scenarios (‘Cross’ and ‘Curve’).

The empirical results demonstrate a clear trend:

1. No Angle Loss (Weight 0): This serves as our baseline, with F1@50 of 80.45 and mF1 of 55.82. Performance in ‘Cross’ and ‘Curve’ scenarios is relatively poor.

2. Low Weight (10): A small weight shows improvement across all metrics, with F1@50 increasing to 80.62 and mF1 to 55.95. There’s also notable improvement in the ‘Cross’ scenario.

3. Optimal Weight (15): This configuration yields the best results across all metrics. F1@75 improves by 0.87%, F1@50 by 0.25%, and mF1 by 0.71% compared to the baseline. Crucially, performance in challenging scenarios improves significantly, with a 24.64% reduction in errors for ‘Cross’ scenarios and a 3.17% improvement in ‘Curve’ scenarios.

4. High Weight (20): While still outperforming the baseline, this configuration shows decreased performance compared to the optimal weight, suggesting that the Angle Loss is beginning to dominate other important loss components.

5. Very High Weight (25): This configuration shows a decline in performance across most metrics, even falling below the baseline in some cases. This indicates that overemphasis on the Angle Loss can be detrimental to overall performance.

These results highlight the critical role of properly calibrated Angle Loss in enhancing model robustness and precision, particularly in complex traffic environments. The optimal weight of 15 strikes a balance between guiding the model to learn angular relationships between lane points and allowing other loss components to contribute effectively.

The substantial improvements in challenging scenarios (‘Cross’ and ‘Curve’) with the optimal weight setting demonstrate the Angle Loss’s effectiveness in helping the model handle complex road geometries. This underscores the potential of finely tuned Angle Loss to significantly enhance overall system effectiveness in real-world driving conditions.

V. CONCLUSION

This research introduces the Global Semantic Enhancement Network (GSENet), a novel architecture designed to address lane detection challenges in complex scenarios.

By integrating the Global feature Extraction Module (GEM) and Top Layer Auxiliary Module (TLAM), GSENet enhances the extraction of global semantic features. We further developed two innovative loss functions: the Angle Loss and the Generalized Line Intersection over Union (GLIoU) Loss, specifically tailored for complex detection environments. Our comprehensive theoretical analyses demonstrate the TLAM module's effectiveness in handling intricate lane detection scenarios and the GLIoU Loss's superiority in optimizing lane point predictions while maintaining global consistency. Experimental validation on the widely recognized CULane and TuSimple benchmark datasets confirms GSENet's significant performance improvements over existing methods, establishing new standards in the field. This superior performance can be attributed to the sophisticated handling of global semantic information and the strategic application of our novel loss functions, enabling more accurate and reliable lane detection across diverse conditions.

REFERENCES

[1] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[2] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI-1, no. 6, pp. 679–698, Nov. 1986.

[3] P. V. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 654 069, Dec. 18, 1962.

[4] I. Sobel and G. Feldman, "A 3×3 isotropic gradient operator for image processing," in *Pattern Classification and Scene Analysis*, 1968, pp. 271–272.

[5] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[6] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–12.

[7] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "RESA: Recurrent feature-shift aggregator for lane detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3547–3554.

[8] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai, and X. He, "CLRNet: Cross layer refinement network for lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 888–897.

[9] Z. Qin, W. Huanyu, and X. Li, "Ultra fast structure-aware deep lane detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 276–291.

[10] Z. Wang, W. Ren, and Q. Qiu, "LaneNet: Real-time lane detection networks for autonomous driving," 2018, *arXiv:1807.01726*.

[11] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "CurveLane-NAS: Unifying lane-sensitive architecture search and adaptive point blending," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 689–704.

[12] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3753–3762.

[13] Z. Qin, P. Zhang, and X. Li, "Ultra fast deep lane detection with hybrid anchor driven ordinal classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 1, pp. 1–14, Jul. 2022.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Aug. 2016, pp. 779–788.

[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[17] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[18] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6–7, Feb. 2014.

[19] X. Li, J. Li, X. Hu, and J. Yang, "Line-CNN: End-to-end traffic line detection with line proposal unit," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 248–258, Jan. 2020.

[20] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: Real-time attention-guided lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 294–302.

[21] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "PolyLaneNet: Lane estimation via deep polynomial regression," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6150–6156.

[22] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3693–3701.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[24] L. Yang, R. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 11863–11874.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–28.

[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[28] B. Li, Y. Hu, X. Nie, C. Han, X. Jiang, T. Guo, and L. Liu, "DropKey for vision transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22700–22709.

[29] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

SUYANG XI is a Junior Student with Xiamen University Malaysia, Consistently, he received the national-level scholarships throughout the academic term and achieved full marks in mathematical subjects, including advanced mathematics and linear algebra. He has research cooperation with professors from renowned institutions, such as the University of California at Berkeley and Imperial College London. His research interests include unmanned driving and wireless communications, with certain insights in the field of computer vision. His commitment exists to contribute to the AI era, particularly in areas, such as image segmentation and image analysis in the future.

ZIHAN LIU is currently a Junior Student with Xiamen University Malaysia. He received outstanding academic scholarships for many semesters and obtained full marks in professional subjects, such as linear algebra, engineering physics, and circuits and equipment. He has interned with a large logistics and transportation company and participated in optimizing logistics information systems and strengthening the warehouse management system (WMS) and other work. During this period, technical tools are used to manage the e-commerce warehouse, including inventory monitoring, order processing, and shipping operations, which greatly improves hands-on ability and professional practical ability. His main research interests include digital design, signal processing, and embedded system design.

ZIMING WANG is a Junior with Xiamen University Malaysia. He has ranked in the top 25% of the department. He has research cooperation with professors with New York University and other universities. His main research interests include solar photovoltaic power generation and embedded systems.

QIANG ZHANG is a Junior, studying electronic information engineering with Zhengzhou University, Henan. During an internship, he has collaborated with a team to develop and design an automatic obstacle avoidance balance vehicle, showcasing strong hands-on ability. He has extensive experience in research and development and design. He demonstrates a lifelong passion for product creation, adept at learning and communication, and applying knowledge to solve problems in work.

HONG DING is a Junior Student with Xiamen University Malaysia, excels in advanced circuits and devices, engineering graphics, and linear algebra, and achieving full marks. He demonstrates outstanding performance in microcircuits, including digital electronics, digital system design, and semiconductor physics. His primary research interests include embedded development and logic circuits, dedicated to contributing to the future connection between digital circuits, and artificial intelligence.

CHIA CHAO KANG (Senior Member, IEEE) received the bachelor's degree from Newcastle University, and the master's and Ph.D. degrees in science from the University of Malaya. He is currently a Senior Lecturer and the Master's Supervisor with Xiamen University Malaysia. He has published papers in the Web of Science and Scopus core collections, with over 20 contributions. With an extensive industrial and teaching experience, his research interests include renewable energy, antennas/wireless, and power systems. He is a Professional Member in Malaysia and a member of the Malaysian Board of Engineers.

ZHENGHAN CHEN is currently a Kaggle Master and a full time employee doing research on Windows Copilot with Microsoft. His current research interests include LLM and AiGent.